Tracking and Data System Near-Earth Telemetry Automatic Switching Unit

L. Butcher
DSIF Operations Section

A hardware–software system is described that is capable of selecting the one from among as many as six incoming data streams which is best by an externally programmed criterion and switching it automatically to the Deep Space Instrumentation Facility (DSIF) Telemetry System. The system has been implemented at the Cape Kennedy Compatibility Test Station to provide the best spacecraft telemetry stream to the DSIF Telemetry System during the near-earth phase of a tracking mission, when as many as six Air Force Eastern Test Range stations are receiving spacecraft telemetry.

I. Introduction

The requirement of the Tracking and Data System for near-earth telemetry support is an important function in the unmanned spacecraft launch support. The requirement is placed upon the Air Force Eastern Test Range and Kennedy Space Center stations to provide spacecraft telemetry in a serial bit stream to DSS 71 after launch until DSN acquisition. The spacecraft telemetry serial bit stream is received at DSS 71 via 202 data modems. It is processed at DSS 71 by the DSIF Telemetry System and transmitted to the SFOF in real-time. There are as many as six near-earth stations receiving and, subsequently, transmitting telemetry data to DSS 71 during the launch phase. The automatic switching unit at DSS 71 provides a means of selecting the best available data and performs automatic switching to select the station data source to output the data to the DSIF Telemetry System for processing. This is to be accomplished with a minimum loss of spacecraft data during the time of launch until loss of signal by the last near-earth supporting station. This data is to be processed at DSS 71 and transmitted to the SFOF in real-time via the GCF high-speed data system.

II. Technical Description

The automatic switching unit consists of an XDS CF-16 system controller and an interface buffer implemented with JPL Hi-Rel digital modules as utilized in the DSIF

Telemetry System. The CF-16 system controller is a packaged unit consisting of digital logic and stored programs with an 8k core memory. Peripheral equipment includes an ASR-33 teletype with paper tape reader and punch for input/output, and an FR-1200 magnetic tape recorder for recording all incoming data lines. A block diagram of the automatic switching unit is shown in Fig. 1.

The interface buffer logic consists of 15 digital Hi-Rel circuit modules mounted in a single chassis for input, output, and data switching to the CF-16 system controller. The telemetry data input from the 202 data modems is ±6-V amplitude. All six inputs are input to negative-to-positive modules for uni-polar output clamped at 5 V. All six lines are then fed to the CF-16 controller to be used for frame sync and line selection when in the automatic mode. Each input data line also is routed through another negative-to-positive circuit to the line selection logic, which has the capacity of selecting one data line whose data is sent to the DSIF Telemetry System for processing.

The data selection can be achieved by either manual or automatic modes. The manual or automatic mode is selectable by a toggle switch on the front panel. When the manual mode is selected, it enables the use of six front panel pushbutton switches to select the desired data source. When the automatic mode is selected, the CF-16 is utilized under program control to select the data source. The output data is also present on an interrupt line and is used for bit sync detection.

A timing signal of 1 kHz is supplied to the CF-16 on an interrupt line. A lift-off pulse or first motion pulse is supplied to the CF-16 to provide time from lift-off for the automatic operation.

III. Functional Description

As presently implemented, the system has a capability of monitoring six input data lines and selecting one line for output data. Criteria for the selection of a data line are data quality, priority of the data, and tracking predict times for each data source.

Data quality is determined by checking for a frame sync pattern in the received data. Counts are kept of the number of good and bad frame syncs for each data line, and therefore a current frame sync status of each line is maintained.

Data line priorities are used to help determine which source to select if two or more lines have a good frame sync. Tracking predict times are used to select a data source if no lines have a recognizable frame sync pattern. Predict times are in seconds from first motion to acquisition of signal (AOS) and loss of signal (LOS) for each station and are counted with a 1-pps clock in the software.

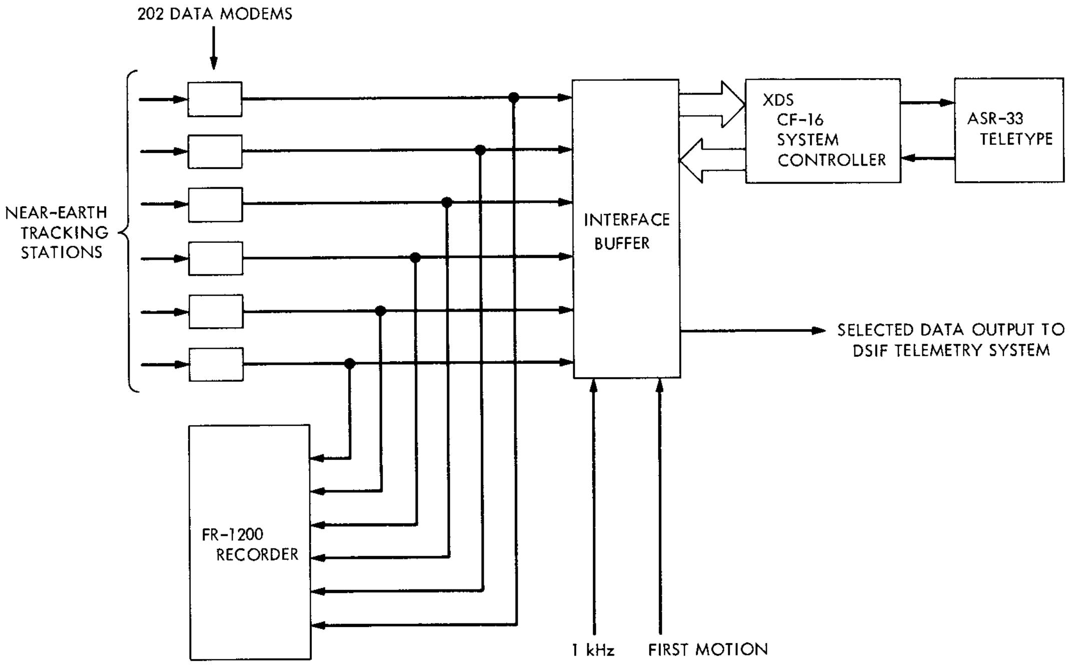
202 DATA MODEMS
NEAR-EARTH TRACKING STATIONS
XDS CF-16 SYSTEM CONTROLLER
ASR-33 TELETYPE
INTERFACE BUFFER
SELECTED DATA OUTPUT TO DSIF TELEMETRY SYSTEM
FR-1200 RECORDER
1 kHz
FIRST MOTION

Fig. 1. Block diagram of near-earth telemetry switching unit

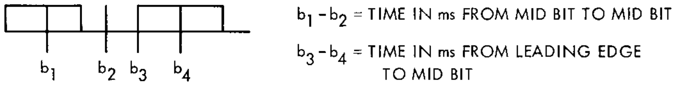
b_1
b_2
b_3
b_4
$b_1 - b_2$ = TIME IN ms FROM MID BIT TO MID BIT
$b_3 - b_4$ = TIME IN ms FROM LEADING EDGE TO MID BIT

Fig. 2. Time references

The software for the system is written entirely in symbolic language and is divided into three sections: a main routine, and two interruptable subroutines. The main routine handles all inputs/outputs and, in addition, performs the line switching for the system. The 1-kHz interrupt subroutine controls all the timing functions for the system and maintains bit sync and frame sync status for each of the six input lines. By determining middle bit time, the data interrupt subroutine is used to help maintain bit synchronization for the 1-kHz subroutine (Fig. 2).

IV. Theory of Operation

A. Main Subroutine

The sequence of operations begins by first setting up the interface logic to accept data on line 1. This would normally be pre-launch and launch data. Then the processor enters an input routine whereby pertinent constants are accepted into storage through the ASR-33 teletype input. These constants are:

(1) Bit rate of expected data.

(2) Lead time in seconds before AOS to begin looking for data.

(3) Number of sequential good sync codes to recognize before setting the frame sync flag.

(4) Number of sequential bad syncs to have in order to reset the frame sync flag.

(5) AOS times for each data line.

(6) LOS times for each data line.

(7) Priority number for the line.

The processor can handle bit rates of either 8⅓ or 33⅓ bps, and is initialized for 33⅓ bps. If 8⅓ bps is entered for the bit rate, the millisecond counts for middle bit time and total bit time are changed accordingly.

After the input routine is completed, the next step in operations is to initialize the message typeout routine and the count for the 1-pps timing. Then the two interrupt routines are enabled and the processor will go into a polling loop to check sync status, AOS, and LOS for each data line.

In the polling loop, the sync status for each line is checked. If only one frame sync is found, the processor will select the data line corresponding to the frame sync. If that line happens to be the one already selected, the re-selection does not affect anything. If more than one frame sync is found, the processor checks the priority of the lines. A line with frame sync and a priority of one takes precedence, but if no line had a priority of one, the first line found with frame sync status would be the selected line. If no lines had a frame sync, the AOS and LOS times would be examined for current time (X) to be in the range $AOS^{(N)} < X < LOS^{(N)}$ where N is the line number. The first data source to meet this condition would be the selected line.

After scanning the status of all the data lines, the next step in the polling loop is a check for launch vehicle first motion. This event, when it occurs, signifies the start of the 1-pps timing clock. The main routine outputs a message to show that first motion has occurred. Until the time of first motion, only a line with an AOS time of zero (normally the preselected line) could be selected and the processor would ignore all other lines.

After sensing for first motion, the main routine next enters a loop where it looks for a change in frame sync status on any line. If a data line has had a change of sync status, a message giving the new sync condition, the line number, and the current 1-pps time value will be typed out.

The 1-pps time value is unfortunately typed out as a hexadecimal number instead of a more readable decimal number. This is the result of a time constraint placed upon the main routine in that if it took the time to convert to decimal before typing out, the portion which has to make the line selection could not be executed often enough. When this loop has been completed for all lines, the main routine will return to the line selection where the cycle begins again.

B. 1-kHz Interrupt Subroutine

A 1-kHz timing pulse controls the execution of this routine so that the 1-pps count, data sampling, data bit sync, and total bit count per frame are all controlled by this timing reference.

The subroutine first checks for a first motion flag to have been set. If the first motion has been sensed, a count of 1000 is decremented to generate the 1-pps clock. If first motion had not occurred, the 1-pps clock would be bypassed. If a 1-pps count is generated, the AOS and

LOS times are decremented for each line so that these counts may be used by the main routine for line selection. No frame syncs will be set for any line not having AOS $\leq$ T $<$ LOS.

For each 1-kHz timing pulse, the subroutine decrements a count which determines the appropriate time to sample a data bit. It represents the time in milliseconds from middle bit time to middle bit time or from leading edge to middle bit time reference (Fig. 2). To aid in bit synchronization, this count is controlled by the data interrupt subroutine.

If the count for mid bit has reached zero, the routine will then sample the data lines simultaneously by inputing in parallel and then performing a parallel-to-serial conversion so that the input block is separated into separate lines. Then each stored word of 16 bits corresponding to each data line is shifted down and merged with the new data bit for each line.

After doing this segment, the subroutine will exit, and on the next 1-kHz count, it performs the portion of the subroutine which checks for frame sync. The reason for the two segments is that the total time required to execute the subroutine is greater than the 1-ms interval between interrupt pulses.

With the next 1-kHz pulse after the sampling pulse, the routine enters the segment to check for frame sync. First, it decrements a count representing the total number of bits in a frame for each line and then looks to see if there is frame sync for each line. If there is, the next check is to see if it is time to look for frame sync again. If it is, the routine then compares the stored input data word against the known frame sync pattern both normal and inverted. If it is a good code, the count for bad syncs is reset. If it is not a good code, the number of bad frame syncs for the line is incremented and the number of good counts is reset. The number of good or bad counts considered to be in or out of lock is a keyboard entry in the main routine. If it were not time to look for the frame sync code, the routine would exit and await the next cycle. If the frame sync flag were not set for the line at the time to look for frame sync, the routine would then check AOS and LOS times. This would be the initial condition for any line before it had good data. If the current time (T) corresponded to AOS $<$ T $<$ LOS for that line, then the routine would check the sync codes. If a sync code is matched with the data word, the number of good frame syncs is incremented. The proper number of consecutive good frame syncs would set the frame sync flag for that line.

Each data line has its own frame sync loop and good and bad count status so that input data lines with a time delay of a data bit time or more between corresponding data bits may be handled simultaneously.

C. Data Interrupt Subroutine

This routine is executed by an interrupt pulse caused by the positive-going edge of a data bit of the data source selected. The subroutine functions as a bit synchronizer by resetting a count which represents the time in milliseconds from leading edge of a data bit to middle bit time (Fig. 2). The priority of this interrupt is higher than that of the 1-kHz interrupt, thus always giving a correct midbit count for data sampling.

D. Power Fail/Safe Routine

In the event of a power failure, two high-priority interrupt locations are available to access the power fail/safe routine; one for power down, the other for power up. When entered from a power down interrupt, the routine saves the contents of the program accessible registers and the program counter register, stores the current selected line number, and then halts the processor. When power comes back on, the power up interrupt again accesses the subroutine, all registers are restored, the line is again selected, and the program returns to the point at which power failed.